Some approaches for the parallelization of the QR decomposition of a matrix

Halil Snopce and Azir Aliu

***Abstract*—**In this paper are investigated some methods for parallel computation of the QR decomposition method of matrices. A mathematical approach is based on the method of given's rotation and the method of householder reflection. The mathematical background is followed by the corresponding array which uses systolic approach. In both cases the systolic array is triangular array. On the case of the systolic array based on given's rotation, parallelization continues step by step as it is shown at figures 5 and 6. The output values of figure 5 become the input for figure 6 and vice versa, the output values of figure 6 become the input for figure 5. This kind of iteration is repeated until achieving the convergence.

***Keywords*—**QR decomposition, Parallelization of QR decomosition, Systolic model, Mapping scheme, Systolic array, given's rotations, Householder reflection, computing the orthonormal matrix, computing the upper triangular matrix.

I. Introduction

THE computation of the QR decomposition of a matrix is one of the most important matrix problems that arises in many applications. The domain of its application is very large. It can be used as an important tool in solving different problems in the area of signal processing, image processing, solution of differential equations etc. It can be applied in the solving of simultaneous linear equations. The QR decomposition of a matrix uses the triangularization of the coefficient matrix, followed by the use of back substitution. Most of the QR -decomposition implementations are based on three methods:

1. The Given's rotation method, (also known as Jacobi rotations) used by W. Givens and originally invented by Jacobi
2. The Gram-Schmidt method and
3. The method with the Householder transformations.

The Householder transformation is one of the most computationally efficient methods to compute the QR-decomposition of a matrix. Although the error analysis [10, 26] shows that the Householder transformation outperforms the Given's method under finite precision computation, the QR decomposition of this method is more difficult. Especially, the systolic approach is difficult because we have to find only local connections. On the other hand, by the method of givens rotations, the order of rotations can be changed influencing different rows. Taking this into the consideration, the parallel processing of this method is very appropriate.

There is a lot of research done in the area of the decomposition of matrices as well as the parallel computation of such decompositions. In the case of the SVD decomposition of matrices we are mentioning the methods based on Jacobi rotations [3] and the method based on Hestenes-Jacobi method [7]. Interesting method for evaluation of sparse Jacobians can be found in [20]. These approaches are followed by the systolic arrays for the parallelization of the SVD decomposition [1, 19, 21]. Using the adaptive singular value decomposition can be found in [24]. The spectral Decomposition of some tridiagonal matrices one can found at [9, 17].

In this paper we analyze the case of the QR decomposition method. The basic idea of the QR-decomposition of a matrix is to express a given $m \times n$ matrix A in the form $A = QR$, where Q is an orthonormal $m \times n$ matrix and R is an $n \times n$ upper triangular matrix with nonzero diagonal entries.

A parallel version of Given's rotation was proposed in [16]. In [14] one can find an alternative way for parallelization of Given's rotation which is more efficient for larger matrices. In [8] it is given a parallel pipeline version of Given's rotation for thr QR decomposition. In [2] one can find the block version of the QR decomposition, which first transforms the matrix into the Hassenberg form and then applies Given's rotation to it. In [11] one can find the design based on Householder method. In [23] is presented a new algorithm for finding QR decomposition for square and full column matrix. The numerical analysis and experiment is given in [15]. In [6] is demonstrated a parallel algorithm based on the Gram-Schmidt method.

The analysis in this paper uses the givens rotation method [12] and the householder method [25]. In [12, 22] are proposed two systolic arrays for the QR decomposition with hardware complexity $O(n^2)$ and time complexity O(n) which are based on the method of Given's rotation. The systolic approach based on the same method can be found at [18].

In this paper we give the mathematical backround of the QR decomposition method, and then we analyze the corresponding parallelization for processing with this method.

II. The QR decomposition Based on Given's Rotation

The upper triangular matrix is obtained using sequences of Given's rotations [3] such that the subdiagonal elements of the first column are nullified first, followed by those of the second column and so forth, until an upper triangular matrix is reached. The procedure can be written in the form given below:

$$Q^T A = R$$

where $Q^T = Q_{n-1}Q_{n-2}...Q_1$ (1)

and $Q_p = Q^{p,p}Q^{p+1,p}...Q^{n-1,p}$

where $Q^{p,q}$ is the Given's rotation operator used to annihilate the matrix element located at row $q+1$ and column p. When we work with 2x2 matrices, an elementary Given's transformation has the form:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} 0 & . & . & . & 0 & r_i & r_{i+1} & . & . & . & r_k \\ 0 & . & . & . & 0 & x_i & x_{i+1} & . & . & . & x_k \end{bmatrix} = \begin{bmatrix} 0 & . & . & . & 0 & r_i' & r_{i+1}' & . & . & . & r_k' \\ 0 & . & . & . & 0 & 0 & x_{i+1}' & . & . & . & x_k' \end{bmatrix} \quad (2)$$

where c and s are the cosine and the sine of the annihilation angle, such that

$$c = \frac{r_i}{\sqrt{r_i^2 + x_i^2}}, \qquad s = \frac{x_i}{\sqrt{r_i^2 + x_i^2}}.$$

It is not difficult to verify that the product of two rotations is also rotation. Let A be an nxn matrix. In order to transform A into an upper triangular matrix R, we can find a product of rotations $Q^T = Q_{n-1}Q_{n-2}...Q_1$ such that $Q^T A = R$. It is not difficult to show that O(n^2) rotations are required. Because the number of operations in every rotation is O(n), the complexity of this algorithm will be O(n^3). In general, the computational complexity of the QR decomposition is given below [6].

1. Householder: 4/3n^3+ O(n^2)
2. Given's: 8/3n^3+ O(n^2)
3. Fast Given's: 4/3n^3+ O(n^2)
4. Gram-Shmidt: 2n^3+ O(n^2)

From the results above it is not difficult to conclude that the Householder transformation outperforms the Given's method under finite precision computation. But on the other hand due to the vector processing nature of the Householder transformation, no local connections in the implementation of the array are necessary. Therefore QR decomposition by the method of Householder transformation is more difficult.

III. Triangular Systolic Array Based on Given's Rotation

In [5] it is shown that a triangular systolic array can be used to obtain the upper triangular matrix R based on sequences of Given's rotations. This systolic array is shown in Fig. 1.

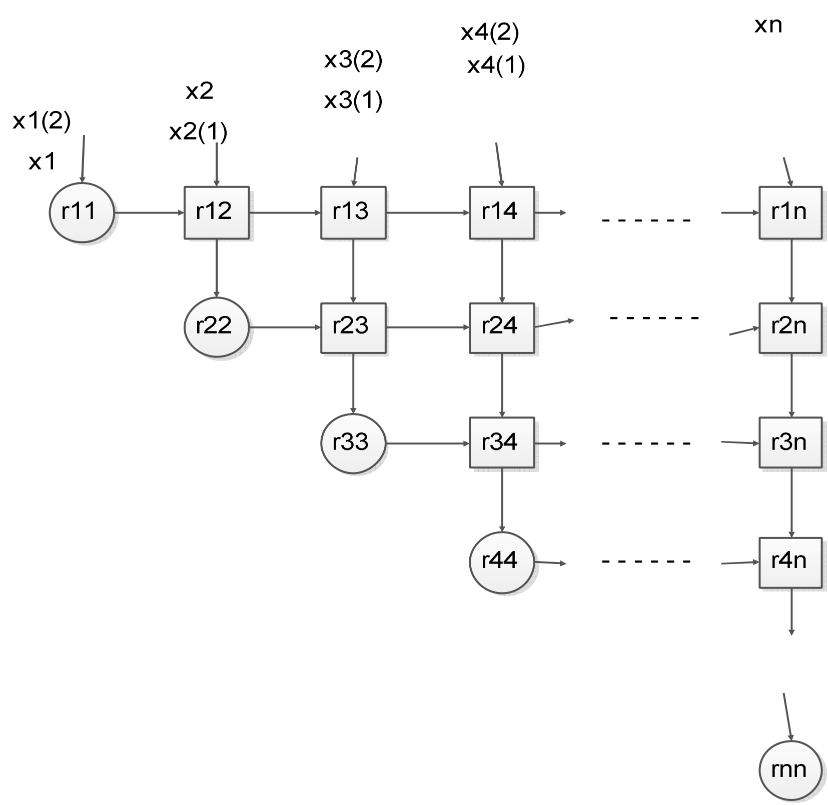

Fig. 1 Triangular systolic array for computing the upper triangular matrix R

As we can see, the array consists of two different shapes of cells. The cells in the shape of a circle (fig. 2), and the cells in quadratic shape (as in fig.3).

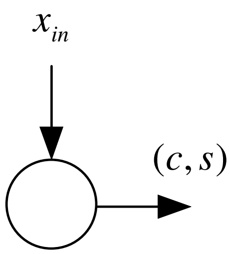

Fig. 2 Input and output of the circle cell of the array in fig.1

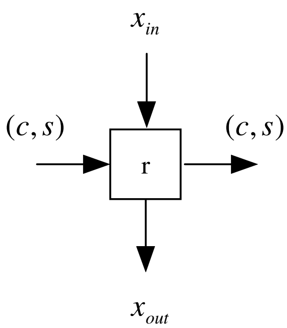

Fig. 3 Input and output of the quadratic cell of the array in fig.1

The cells of fig.2 perform according to algorithm 1:

Algorithm 1:

$$
\begin{aligned}
& \textit{If } \; x_{in} = 0 \; \textit{ then} \\
& \quad c = 1; \; s = 0 \\
& \textit{otherwise} \\
& \quad r' = \sqrt{r^2 + x_{in}^2}; \\
& \quad c = r/r'; \; s = x_{in}/r' \\
& \quad r = r'; \\
& \textit{end}
\end{aligned}
$$

The calculations of quadratic cells are given by the relations below:

$$
\begin{aligned}
x_{out} &= c x_{in} - s r \\
r &= s x_{in} + c r
\end{aligned}
$$

According to the relations in (1), the Q matrix cannot be obtained by multiplying cumulatively the rotation parameters propagated to the right. Accumulation of the rotation parameters is possible by using an additional rectangular systolic array.

Before giving an explanation about the systolic array for the QR decomposition, we will introduce the methodology of computing $R^{-T}x$. This computation will be used in the general design of the systolic array for a QR decomposition.

IV. The Computation of $R^{-T}X$

We present a brief derivation of the result presented in [13] about the property that a triangular array can compute $R^{-T}x$ in one phase with the matrix R situated in that array.

Let $r_{ij} = [R]_{ij}$ and $r'_{ij} = [R^{-1}]$, where $r_{ij} = 0$ and $r'_{ij} = 0$ for $i > j$. It can be shown that:

$$
r'_{ij} = \begin{cases} \dfrac{1}{r_{ii}}; & i = j \\ -\displaystyle\sum_{k=i}^{j-1} \dfrac{r'_{ik} r_{kj}}{r_{jj}}; & i < j \le n \end{cases} \tag{3}
$$

Let

$$
[y_1, \; . \; . \; ., \; y_n]^T = R^{-T} X \tag{4}
$$

Then the recursive computation of (4), where R^{-T} is a nxn matrix and X is an nxm matrix is:

$$
y_j = \sum_{i=1}^{j} x_i r'_{ij}, \qquad i = 1, \; . \; . \; ., \; n \tag{5}
$$

In particular (because we want to use R and X to compute $R^{-T}X$), y_j can be expressed in terms of r_{ij} and x_i. By substituing the equation (4) into equation (5) we have:

$$
\begin{aligned}
y_j &= \sum_{i=1}^{j} x_i r'_{ij} = y_j = \sum_{i=1}^{j-1} x_i r'_{ij} + x_j r'_{jj} = \\
&= \sum_{i=1}^{j-1} x_i r'_{ij} + \frac{x_j}{r_{jj}}
\end{aligned} \tag{6}
$$

If we continue, by transforming the relation (6), we will have:

$$
y_j = \frac{x_j}{r_{jj}} + \sum_{i=1}^{j-1} x_i r'_{ij} = \frac{x_j}{r_{jj}} - \sum_{i=1}^{j-1} x_i \sum_{k=i}^{j-1} \frac{r'_{ik} r_{kj}}{r_{jj}}
$$

And finally we get:

$$y_j = \frac{1}{r_{jj}} \cdot \left(x_j - \sum_{i=1}^{j-1} x_i \sum_{k=i}^{j-1} r'_{ik} r_{kj} \right) = \\ = \frac{1}{r_{jj}} \cdot \left(x_j - \sum_{k=1}^{j-1} \sum_{i=1}^{k} x_i r'_{ik} r_{kj} \right) \quad (7)$$

Using the relation (5), for the final form of y_j, we get:

$$y_j = \frac{1}{r_{jj}} \left(x_j - \sum_{k=1}^{j-1} y_k r_{kj} \right) \quad (8)$$

Finally, using the relations obtained above (where Y is the nxm matrix, R is nxn upper triangular matrix and X is an nxm matrix), the algorithm for computing $R^{-T}x$ is given:

Algorithm 2

for $i = 1$ *to* n

$y_1 = \frac{1}{r_{11}} \cdot x_1$

for $j = 2$ *to* n

begin

$z_j = x_j$

for $k = 1$ *to* $j-1$

$z_j = z_j - y_k r_{kj}$

$y_j = \frac{z_j}{r_{jj}}$

end

The corresponding systolic array is similar as the array in fig.1. The data movement of input values x and output values y is presented in the figure 4.

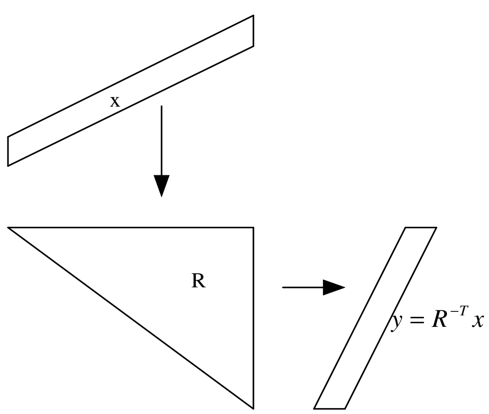

Fig. 4 Data movement of x and y in the computation of $R^{-T}x$

In the case presented above, the elements of the matrix R are stored in the triangular array. The cells of fig.2 (circle cells) perform the division part of the equation (8) (the part $1/r_{jj}$). The second part of the eq. (8) (the part $x_j - \sum_{k=1}^{j-1} y_k r_{kj}$) is performed by the quadratic cells shown in fig.3.

V. Mapping into the Systolic Array

The number of processors in the fig. 1 can be given by the formula $\frac{p(p+1)}{2}$ for some integer number p. To give the mapping scheme of this array we assume that $m = \left\lfloor \frac{n}{p} \right\rfloor$ and that the cell on the position (i, j) is mapped on to the processor (s, k) in the corresponding network. The mapping is given by the formula [4]:

$$s = \begin{cases} \left\lfloor \frac{i}{m+1} \right\rfloor & if \quad \frac{i}{m+1} < n \bmod p \\ \left\lfloor \frac{i - n \bmod p}{m} \right\rfloor & otherwise \end{cases}$$

And

$$k = \begin{cases} \left\lfloor \frac{j}{m+1} \right\rfloor & if \quad \frac{j}{m+1} < n \bmod p \\ \left\lfloor \frac{j - n \bmod p}{m} \right\rfloor & otherwise \end{cases}$$

The relations given above produces a uniform mapping in the case when p is divisible with n. On the other hand (when p doesn't divide n), some processors (in the first $\boldsymbol{n \bmod p}$ columns and $\boldsymbol{n \bmod p}$ rows), take a matrix which is one dimension larger.

VI. The QR Systolic Array

The design of the systolic array for a QR-decomposition of a matrix A will be based on an iterative algorithm which consists of two basic steps. Initially we set $A_1 = A$. The first step is to compute $A_k = Q_k R_k$. The process has to be continued until the convergence. To compute the next iteration A_{k+1} we start from the relation (1) and taking into the consideration that Q is orthonormal ($Q^T Q = I$), we have:

$$Q_k^T A_k = R_k \Rightarrow A_k = Q_k R_k \Rightarrow A_k R_k^{-1} = Q_k \quad (9)$$

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k = Q_k^T Q_k R_k Q_k = R_k Q_k \quad (10)$$

So, this can be expressed as follows:

Algorithm 3

Set $A = A_1$
Step 1: For $k = 1, 2, \ldots,$ *compute* $A_k = Q_k R_k$.

Step 2: Compute $A_{k+1} = R_k Q_k$. *If* A_{k+1} *converges, then stop. Otherwise go back to step 1.*

From $A_k = Q_k R_k$ we have that

$$A_k^{\ T} = R_k^{\ T} Q_k^{\ T} \Rightarrow R_k^{-T} A_k^{\ T} = Q_k^{\ T}$$

If the i th column of the matrices $A_k^{\ T}$ and $Q_k^{\ T}$ is denoted by a_i and q_i respectively, then:

$$R_k^{-T}\left[a_1 \quad a_2 \quad . \quad . \quad . \quad a_n\right] = \\ = \left[q_1, \quad q_2, \quad . \quad . \quad . \quad q_n\right] \tag{11}$$

We already have shown how to compute $R^{-T}x$. So, the systolic array is similar to that one shown in fig. 4. Since the i -th column of $A_k^{\ T}$ is the same with the i-th row of A_k, the elements of the matrix A_k will be inputted row by row. The corresponding systolic array for computing the elements of Q_k as output elements is given in fig. 5.

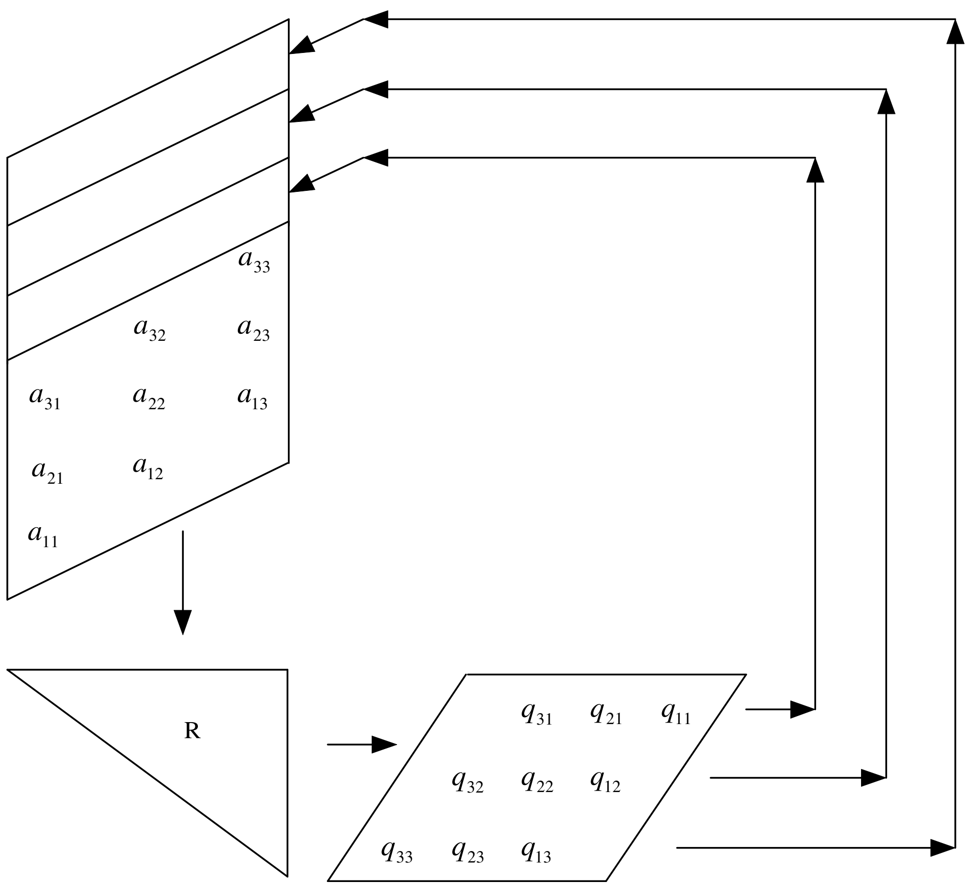

Fig. 5 Systolic model for computing the Q matrix

Of course, the triangular array which contains the elements of the matrix R (presented as right angle triangle) is the same with the array in fig.1.

Fig. 5 in fact is the design for systolic computing of step 1 of algorithm 2. To do the second step, which consists in computing $A_{k+1} = R_k Q_k$, the output elements of fig.5 become row by row the input elements for the new computation. It is illustratively shown in fig. 5. So, the new array, which computes the element of the matrix A_{k+1} using as an input elements the computed ones illustrated in fig. 5, is shown in fig. 6. The elements of A_{k+1} come out column by column.

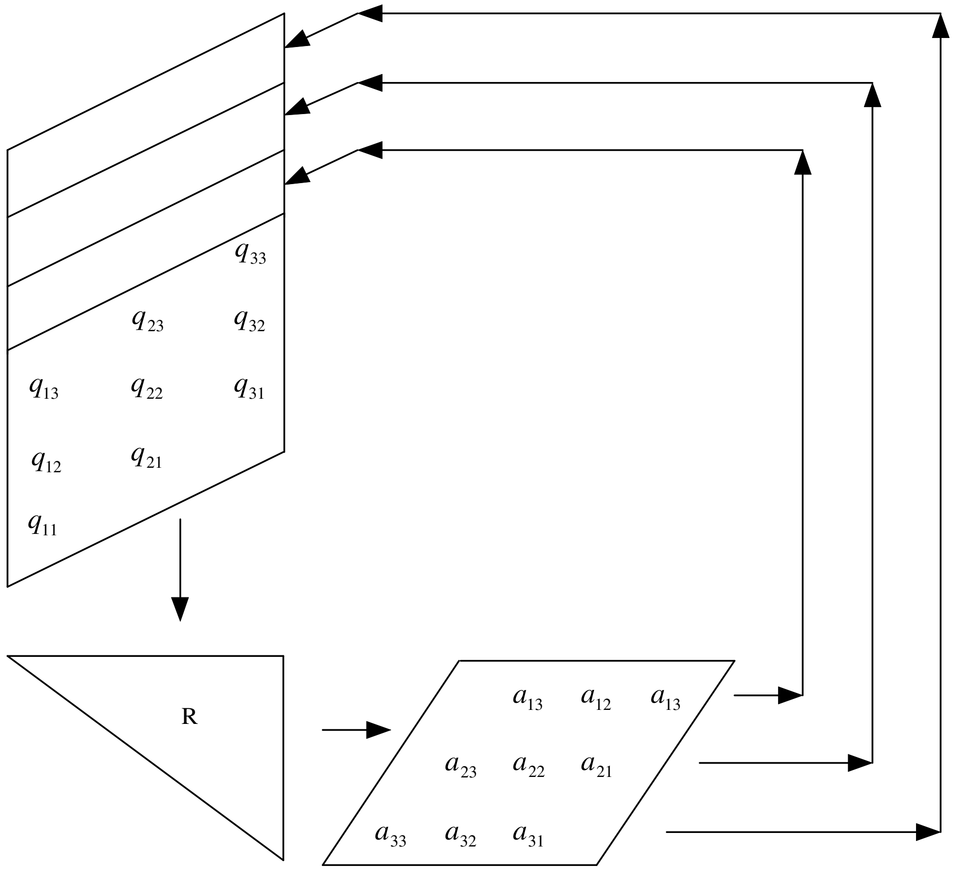

Fig. 6 Systolic computing of the product RQ

If the obtained result is not convergent, then a new iteration will be repeated until achieving a convergence.

VII. Parallel Algorithm Based on Householder Reflections

Let's take the matrix $A = I - \tau u u^T$ where $u \neq 0$ and τ is a constant which is not equal to 0. The purpose is to choose τ such that A is orthogonal ($A^T A = I$). We have:

$$\begin{aligned} A^T A &= (I - \tau u u^T)^T (I - \tau u u^T) \\ &= I - 2\tau u u^T + \tau^2 u u^T u u^T \\ &= I - 2\tau u u^T + \tau^2 (u^T u) u u^T \\ &= I + (\tau^2 u^T u - 2\tau) u u^T \\ &= I + \tau(\tau u^T u - 2) u u^T \end{aligned}$$

From above, if $\tau = 2/u^T u$, then $A^T A = I$. If we take $u^T u = 1$ then $A = I - 2vv^T$, where $v^T v = 1$.

Householder reflection first implements the decomposition:

$$Q_1 A = R = \begin{bmatrix} x & x & x & \cdots & x \\ 0 & & & & \\ 0 & & A_k & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix}$$

where $Q_1 = I - 2\frac{uu^T}{u^T u}$ and A_1 is the first vector of A. The matrix Q can be obtained applying the formula $Q^T = Q_{n-1} Q_{n-2} \dots Q_1$.

Graphical representation of the computation of A is given as below:

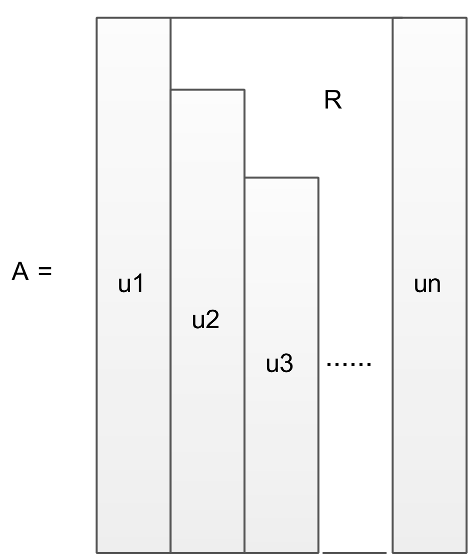

Fig. 7. Graphical representation of the computation of A

The corresponding algorithm is given as in below [25]:

Algorithm 4

$for \;\; j = 1 \; to \; n \; do$

$s = 0$

$for \; i = j \; to \; m \; do \;\; s = s + a_{ij}^2$

$s = sqrt(s); d_j = -s \, if \;\; a_{jj} > 0, \; else \; d_j = s$

$F = sqrt(s * (s + abs(a_{jj})));$

$a_{jj} = a_{jj} - d_j;$

$for \; k = j \; to \; m \; do \; a_{kj} = a_{kj} / F;$

$for \; i = j + 1 \; to \; n$

$begin$

$s = 0;$

$for \; k = j \; to \; m \; do \; s = s + a_{kj} * a_{ki};$

$for \; k = j \; to \; m \; do \; a_{ki} = a_{ki} - a_{kj} * s;$

end

end

The dependence graph and the corresponding array using the projection direction [1 0 0] are given in the fig. 8 and fig. 9.

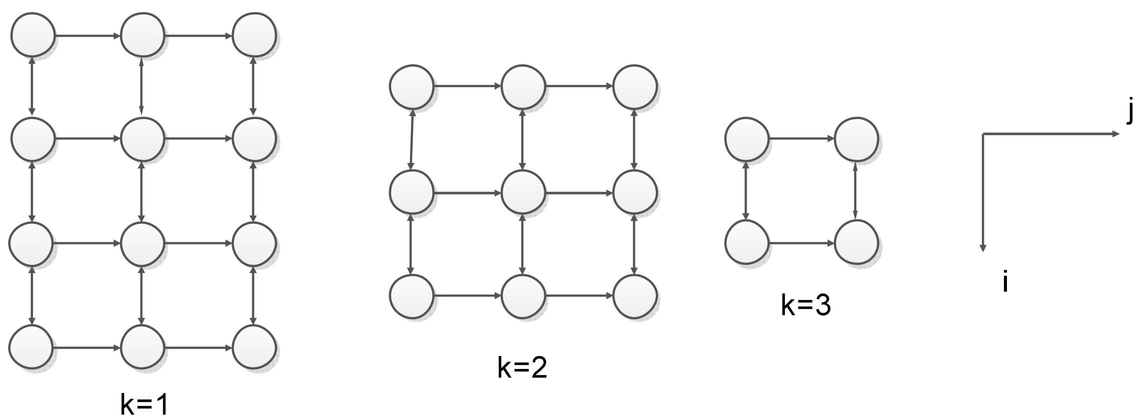

Fig. 8. Dependence graph of the systolic array for QR decomposition using hauseholder reflections

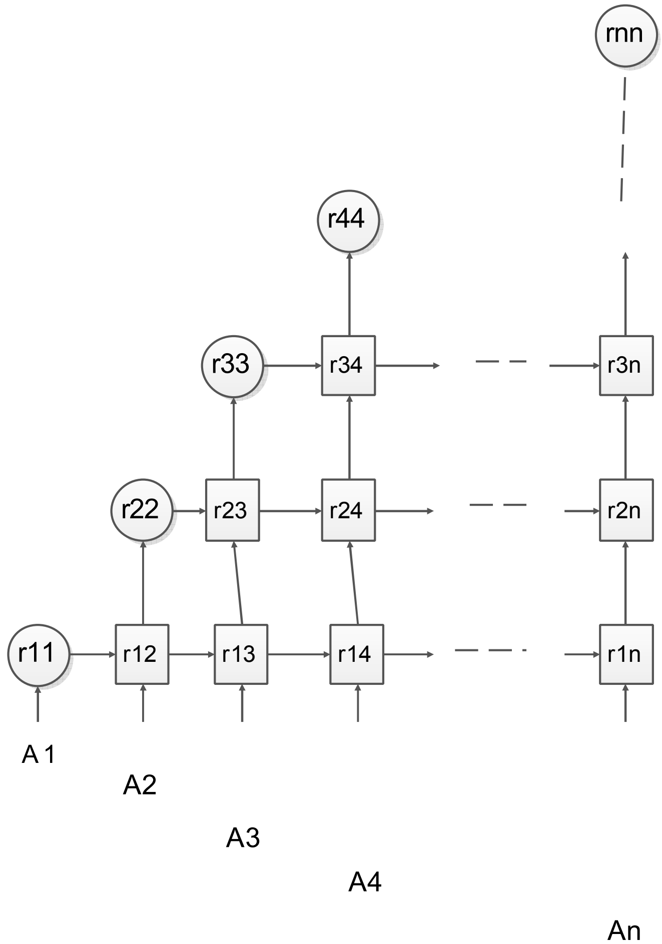

Fig. 9. Systolic array for parallel QR decomposition using hauseholder reflection

In the first step u, v(1) and Q are computed in the first column. Then there is a movement of u and Q in the direction of j axis, and then v(2), ….., v(n) are computed correspondingly in respective columns. In the case of fig. 9, A_i represents the column i of matrix A. As we can see the array is triangular array with the hardware complexity of $O(n^2)$. The array consists of two different shapes of cells. The cells in the shape of a circle (fig. 10), and the cells in quadratic shape (as in fig.11).

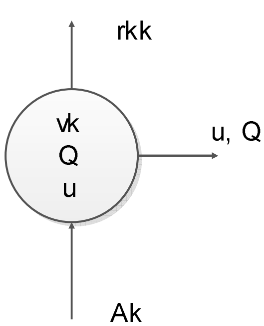

Fig. 10. Input and output of the circle cell of the array in fig.9

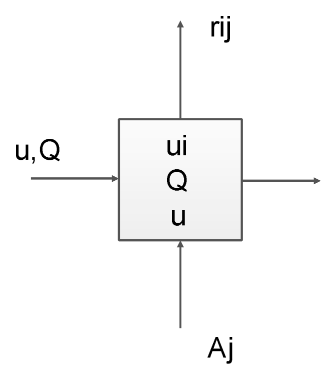

Fig. 11. Input and output of the quadratic cell of the array in fig.9

REFERENCES

[1] Ahmedsaid, A., Amira, A., and Bouridane, A. (2003), Improved SVD systolic array and implementation on FPGA, in: *IEEE International Conference on Field Programmable Technologie*, pp. 3-42.

[2] Berry, M., Dongara, J. and Kim, Y. (1995) 'a parallel algorithm for the reduction of a nonsymmetric matrix to block upper-Hessenberg form,' *Parallel Computing*, vol. 21, no. 8, pp. 1189-1211.

[3] Carl G.J. Jacobi. 'Uber eine neue Auflosungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen Astronomishe Nachricten' 22, 1845. English translation by G.W. Stewart, Technical Report 2877, Department of Computer Science, University of Maryland, April 1992.

[4] D'Cierno, A., Ceccarelli, M., Farina, A., Petrosino, A. and Timmoneri, L. (1994), 'Mapping QR Decomposition on Parallel Computers', *A Study case for Radar applications*, *IEICE Trans. Commun.*, vol. E77-B, No. 10.

[5] Gentleman, W.M. and Kung, H.T. (1981) 'Matrix triangularization by systolic array', *Proc. SPIE Int. Soc. Opt. Eng*., vol. 298, p. 298.

[6] Ghodsi, S.R., Mehri, B. and Taeibi-Rahni, M. (2010), " A parallel implementation of Gram-Schmidt Algorithm for Dense Linear Systems of Equations", *International Journal of Computer Applications*, Vol. 5-No. 7, pp. 16-20.

[7] Hestenes, M. R. (1958), 'Inversion of Matrices by Biorthogonalization and Related Results', *Journal of the Society for Industrial and Applied Mathematics,* 6(1): 51-90.

[8] Hofmann, M. and Kontoghiorghes, E. (2006) 'pipeline Givens Sequences for computing the QR decomposition on a EREW PRAM,' *Parallel computing*, vol. 32, no. 3, pp. 222-230.

[9] Jiang, Zh., Shen, N. and Li, J. (2013), 'The Spectral Decomposition of some Tridiagonal Matrices', *WSEAS Transactions on Mathematics,* pp. 1135-1145, volume 12, issue12.

[10] Johnsson, L., "A Computational Array for the QR Method". *In Proc. 1982 Conf. Advanced Res. VLSI (M.I.T., Cambridge, MA)*, pp.123-129.

[11] Kahaner, D., Moler, C. and Nash, S. (1989) '*Numerical methods and Software*'; Printce Hall, Englewood Cliffs.

[12] Kung, S. Y.(1988) *VLSI array processors*; Prentice Hall, New Jersey.

[13] McWhirter, J.G. and Shepherd, T.J., "An efficient systolic array for MVDR beamforming", *in Proc. Int. Conf.Systolic Array*, 1988, pp.11-20.

[14] Modi, J. and Clarke, M. (1984)'An alterantive givens ordering', *Numerische Matheamtik*, vol. 43, no.1, pp. 83-90.

[15] Nugraha, A. S. and Basaruddin, T., "Analysis and Comparisons of QR Decomposition Algorithm in Some Types of Matrix", *Proc. Of the Federated onference on Computer Science and Information Systems*, 2012, pp. 561-565.

[16] Sameh, A.H. and Kuck, D.J. (1978) 'On stable parallel linear system solvers' J. ACM, vol. 25, pp. 81-95.

[17] Shen, N., Jiang, Zh. and Li, J. (2013), 'The Spectral Decomposition of near-Toeplitz tridiagonal matrices', *International Journal of Applied Mathematics and Informatics,* pp. 115-122, Issue4, volume 7.

[18] Snopce, H. and Aliu, A. 'Systolic Approach for QR Decomposition' (2014), *SPRINGER VERLAG volume with title 'computational problems in engineering II', from the conference AMCSE 2014,* Varna Bulgaria.

[19] Snopce, H., and Spahiu, I. (2010), Parallelization of SVD of a Matrix-Systolic Approach, *International Multiconference on computer Science and Information Technology, pp.* 343-348, Wisla.

[20] Strogies, N. and Griewank, A., Efficient Evaluation of Sparse Jacobians by Matrix Compression: Implementation and Experiments, *WSEAS Transactions on Mathematics,* pp. 646-653, volume 13, 2014.

[21] Sun, C.C. and Gotze, J. 'A VLSI design for parallel iterative algorithms', *Proceedings of the 9th int. conf. on Communications and information technologies*, pp. 688-692, IEEE press Piscataway, NY, USA, 2009.

[22] Swartzlander, E. E. Decker, M. (1987) '*Systolic Signal Processing Systems*'; New York.

[23] Tianxiang, F. and Hongxia, L., "The computer realization of the QR Decomposition on Matrices with Full column rank", *Proc. IEEE int. Conf. On computer Intellegence and Security.* NO (2009), 76-79.

[24] Udomhunsakul, S., Noise (2013), 'Reduction using Adaptive Singular Value Decomposition', *International journal of Circuits, Systems and Signal Processing,* pp. 91-100, volume 7.

[25] Walter, G. (2003), "Seminar Fuer Angewandte Mathematic Eidgenoessische Technische Hochschule", *Research report No. 80-02, CH-8092* Zuerich.

[26] Wilkinson, J.H. (1965) 'The Algebraic Eiginvalue Problem'. Oxford.

Halil Snopce was born in Tetovo (R. Macedonia) in 1973. Received his BS from Faculty of Natural and Mathematical Sciences, University "St. Kiril and Methodius "-Skopje, Macedonia, Master's degree in Mathematical department from the "University of Tirana"-Tirana, Albania, and PhD degree in Applied Computer Science, Faculty of Contemporary Sciences and Technologies, "South East European University "-Tetovo, Macedonia. Currently he is assistant professor in SEE University, faculty of Computer Science and Technologies. His research interest include numerical methods and computation using parallel processing.

Azir Aliu was born in Gostivar (R. Macedonia) in 1976. Received his BS from Faculty of Electrical Engineering, major Computer Science and IT, University "St. Kiril and Methodius "-Skopje, Macedonia, Master's degree in Computer Science from University "St. Kiril and Methodius "-Skopje,and PhD degree in Computer Science, Faculty of Mathematics and Natural Sciences-Department of Informatics, University "St. Kiril and Methodius "-Skopje, Macedonia. He was vice-dean for postgraduate studies in Faculty of Computer Science and Technologies, South East European University-Tetovo, Macedonia and Director of Research Center in CST, coordinator of national international and European research projects. Currently he is assistant professor in SEE University, faculty of Computer Science and Thechnologies and Director of Technology Park SEEUTechPark. His research interest include computation methods in image processing.